# One-step differentiation of iterative algorithms

Toulouse School of Economics — UNIVERSITÉ CÔTE D'AZUR — cnrs

Jérôme Bolte, Edouard Pauwels, Samuel Vaiter

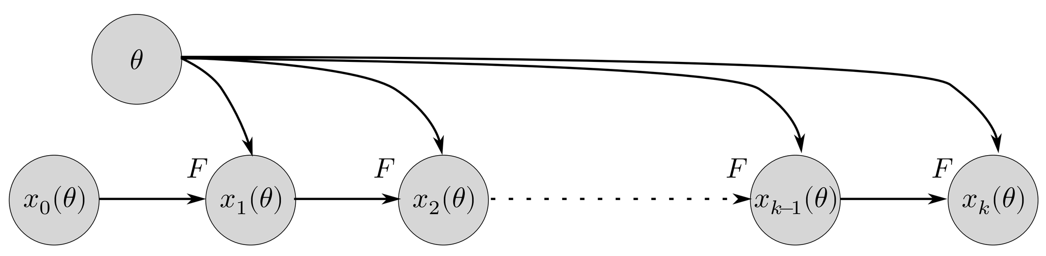
NeurIPS'23 Spotlight

When your algorithm is fast,
it is enough to differentiate **only** the last iterate

## Parametric iterative algorithm

$$\begin{cases} x_0(\theta) \in \mathbb{R}^n \\ x_{k+1}(\theta) = F(x_k(\theta), \theta) \end{cases}$$

$F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$
Recursive map



*ex:* gradient descent,
Newton method,
recurrent architectures,
Deep Equilibrium Network, etc.

## Automatic differentiation

**Input:**
$\theta \mapsto x_0(\theta) \in \mathcal{X}$, $k > 0$.
**Eval:**
`with_gradient`
**for** $i = 1, \dots, k$ **do**
  $x_i(\theta) = F(x_{i-1}(\theta), \theta)$
**Return:** $x_k(\theta)$
**Differentiation:** native autodiff on **Eval**.

## Implicit differentiation

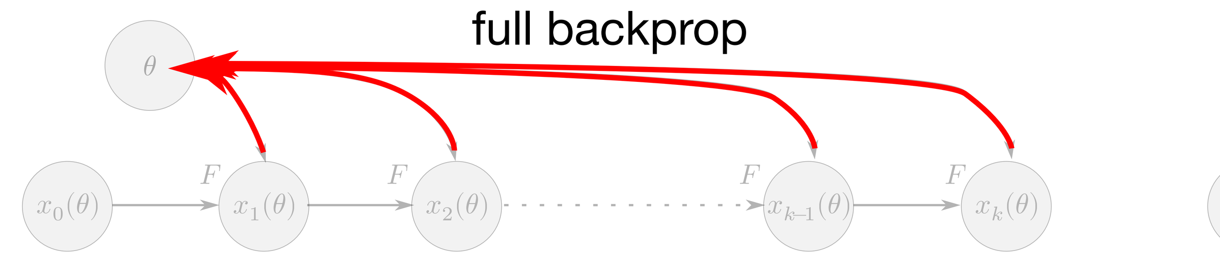
**Input:**
$x_0 \in \mathcal{X}$, $k > 0$.
**Eval:**
`stop_gradient`
**for** $i = 1, \dots, k$ **do**
  $x_i = F(x_{i-1}, \theta)$
**Return:** $x_k$
**Differentiation:** Custom VJP / JVP.

## One-step differentiation

Jacobian-Free Backpropagation

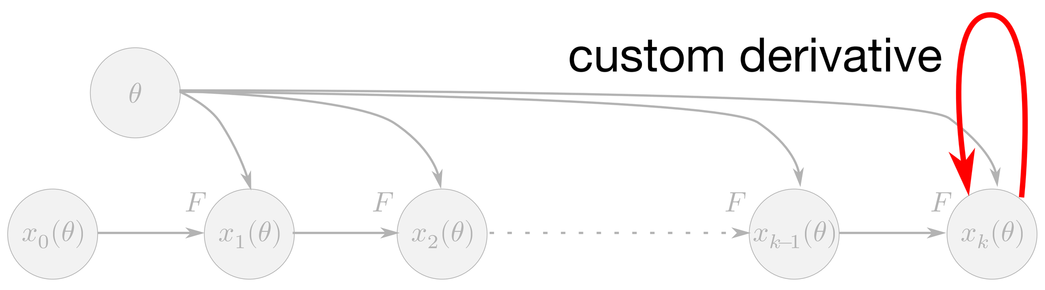
**Input:** $x_0 \in \mathcal{X}$, $k > 0$.
**Eval:**
`stop_gradient`
**for** $i = 1, \dots, k-1$ **do**
  $x_i = F(x_{i-1}, \theta)$
`with_gradient`
$x_k(\theta) = F(x_{k-1}, \theta)$
**Return:** $x_k(\theta)$
**Differentiation:** native autodiff on **Eval**

full backprop



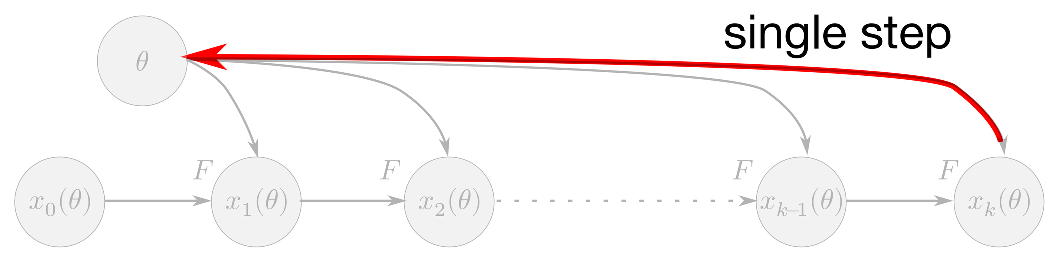
$$J_\theta x_{i+1}(\theta) = J_x F(x_i(\theta), \theta) J_\theta x_i(\theta) + J_\theta F(x_i(\theta), \theta)$$

custom derivative



$$J^{\mathrm{ID}} x_k(\theta) = (I - J_x F(x_k(\theta), \theta))^{-1} J_\theta F(x_k(\theta), \theta)$$

single step



$$J^{\mathrm{OS}} x_k(\theta) = J_\theta F(x_{k-1}(\theta), \theta)$$

## Linearly convergent algorithms

**Ass. (Contract.).** Let $F \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ be $C^1$, $\rho \in [0, 1)$, and $\mathcal{X} \subset \mathbb{R}^n$ be nonempty convex closed, s.t., for any $\theta$, $F_\theta(\mathcal{X}) \subset \mathcal{X}$ and $\|J_x F_\theta\|_{\mathrm{op}} \leq \rho$.

$\overset{\text{loc.}}{\Longleftarrow}$

$\Downarrow$

$x_k(\theta) \to \bar{x}(\theta)$ and the convergence is linear

**Proposition.** *Let $F$ and $\mathcal{X}$ such that $\theta \mapsto F(x, \theta)$ is $L_F$ Lipschitz and $x \mapsto J_\theta F(x, \theta)$ is $L_J$ Lipschitz for all $x \in \mathbb{R}^n$. Then, for all $\theta \in \mathbb{R}^m$,*

$$\|J^{\mathrm{OS}} x_k(\theta) - J_\theta \bar{x}(\theta)\|_{\mathrm{op}} \leq \frac{\rho L_F}{1-\rho} + L_J \|x_{k-1} - \bar{x}(\theta)\|$$

$$\|J^{\mathrm{ID}} x_k(\theta) - J_\theta \bar{x}(\theta)\|_{\mathrm{op}} \leq \frac{L_J L_F}{(1-\rho)^2} \|x_k - \bar{x}(\theta)\| + \frac{L_J}{1-\rho} \|x_k - \bar{x}(\theta)\|$$

## Superlinear algorithms

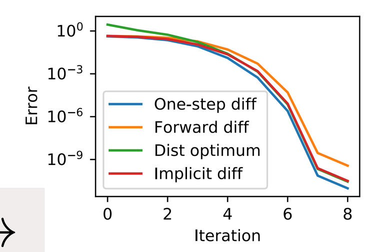
**Ass. (Vanishing Jac.).** Let $F \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ be $C^1$, $x_k(\theta)$ converges globally locally uniformly in $\theta$ to the unique $\bar{x}(\theta)$, and $J_x F(\bar{x}(\theta), \theta) = 0$.

**Proposition** (Jacobian convergence). *Let $F$ satisfying **(Vanishing Jac.)**. Then $J^{\mathrm{OS}} x_k(\theta) \to J_\theta \bar{x}(\theta)$ as $k \to \infty$, and $J^{\mathrm{OS}} \bar{x}(\theta) = J_\theta \bar{x}(\theta)$.*

## Quadratically convergent algorithms

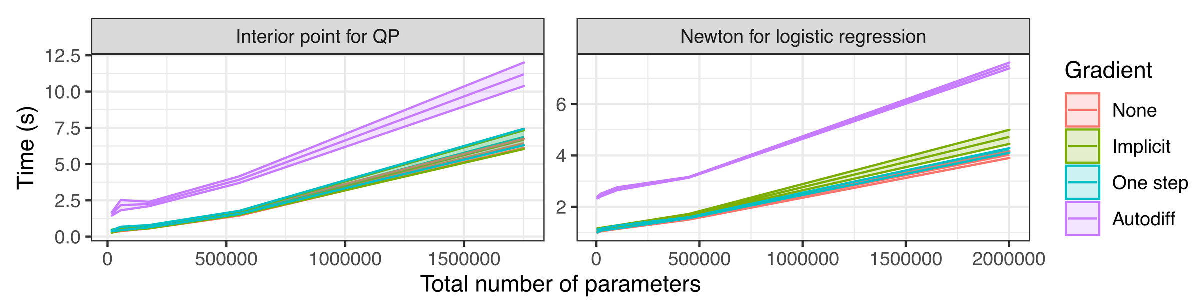
**Proposition.** *Let $F$ satisfying **(Vanishing Jac.)** such that $x \mapsto J_{(x,\theta)} F(x, \theta)$ is $L_J$ Lipschitz. Then,*

$$\|J^{\mathrm{OS}} x_k(\theta) - J_\theta \bar{x}(\theta)\|_{\mathrm{op}} \leq L_J \|x_{k-1} - \bar{x}(\theta)\|.$$

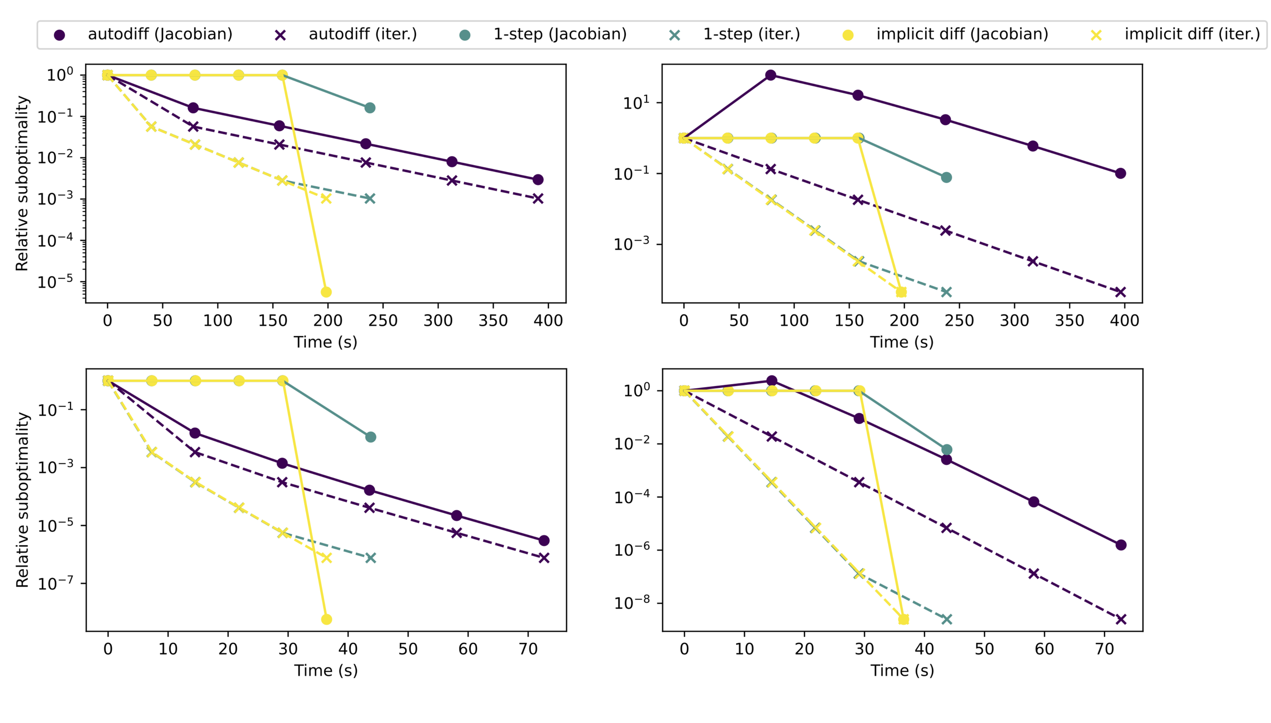



*Newton's quadratic convergence of the derivatives.*

## Numerical illustrations




*Left: timing experiment for differentiable quadratic programs.*
*Right timing experiment for differentiation of Newton algorithm for logistic regression.*
*For Newton experiment, the one step estimator coincide with ID estimator up to $10^{-12}$ error.*
*For the interior point experiment, it coincides with ID estimator up to $10^{-6}$ error.*




*Differentiation of gradient descent for solving weighted Ridge regression on* `cpusmall`.
*Top line: condition number of 1000. Bottom line: condition number of 100.*
*Left column: small learning rate. Right column: big learning rate.*
*Dotted lines: lack of optimality of the iterates. Filled lines: lack of optimality of the Jacobians.*

## Hypergradient descent for bilevel problems

$$\min_\theta g(x(\theta)) \text{ s.t. } x(\theta) \in \arg\min_y f(y, \theta) \quad \Longleftrightarrow \quad \min_\theta g(x(\theta)) \text{ s.t. } x(\theta) = F(x(\theta), \theta)$$

(Hyper-)gradient descent using one-step estimator

$$\theta_{l+1} = \theta_l - \alpha J^{\mathrm{OS}} x_k(\theta_l)^T \nabla g(x_k(\theta_l))$$

**Proposition** (Approximate critical points). *Assume*

- *$F$ satifies **(Contract.)**.*
- *$g$ is $l_g$ Lipschitz and $\nabla g$ is $l_\nabla$ Lipschitz*
- *$\sup_\theta \|x_0(\theta) - F_\theta(x_0(\theta))\| \leq M$, for some $M > 0$.*
- *$F$ is $L_F$ Lipschitz and $J_{(x,\theta)} F$ is $L_J$ Lipschitz jointly.*
- *$g$ is $C^1$, $l_g$ Lipschitz with $l_\nabla$ Lipschitz gradient.*
- *$\frac{1}{\alpha} \geq \left(\frac{L_J}{1-\rho}\left(\frac{L_F}{1-\rho} + 1\right) l_g + l_\nabla \frac{L_F}{1-\rho}\right) \frac{L_F}{1-\rho}$*
- *$g \circ \bar{x}$ is lower bounded by $g^*$.*

*Then setting $\epsilon = \frac{\rho}{1-\rho}(L_F l_g + (L_J l_g + L_F l_\nabla) M \rho^{k-2})$, for all $K$,*

$$\min_{l=0,\dots,K} \|\nabla_\theta (g \circ \bar{x})(\theta_l)\|^2 \leq \epsilon^2 + \frac{2L((g \circ \bar{x})(\theta_0) - g^*)}{K+1}.$$

## References

Bolte, Pauwels, Vaiter.
One-step differentiation of iterative algorithms.
NeurIPS. 2023.

Finn, Abbeel, Levine.
Model-agnostic meta-learning for fast
adaptation of deep networks
ICML. 2017.

Fung, Heaton, Li, McKenzie, Osher, Yin.
Jacobian-free backpropagation for implicit models.
AAAI. 2022.

Geng, Zhang, Bai, Wang, Lin.
On training implicit models.
NeurIPS 2021.

Shaban, Cheng, Hatch, Boots.
Truncated back-propagation for bilevel optimization
AISTATS. 2019.